PCT/KR 03/00992
RO/KR 20.05.2003

대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2003-0023481
Application Number

REC'D 06 JUN 2003
WIPO PCT

출 원 년 월 일 : 2003년 04월 14일
Date of Application APR 14, 2003

출 원 인 : 주식회사 하우리
Applicant(s)

2003 년 04 월 22 일

특 허 청
COMMISSIONER

PRIORITY
DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0005
【제출일자】	2003.04.14
【국제특허분류】	G06F
【발명의 명칭】	메모리를 감염시키는 바이러스의 치료방법, 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체 및 바이러스의 치료장치
【발명의 영문명칭】	Curative Method for Computer Virus Infecting Memory, Recording Medium Comprising Program Readable by Computer, and The Device
【출원인】	
【명칭】	주식회사 하우리
【출원인코드】	1-1998-612859-6
【대리인】	
【성명】	황이남
【대리인코드】	9-1998-000610-1
【포괄위임등록번호】	2000-014588-4
【발명자】	
【성명의 국문표기】	권석철
【성명의 영문표기】	KWON,SEOK CHUL
【주민등록번호】	700322-1470611
【우편번호】	156-020
【주소】	서울특별시 동작구 대방동 502 현대아파트 102동 1501호
【국적】	KR
【발명자】	
【성명의 국문표기】	최원혁
【성명의 영문표기】	CHOI,WON HYOK
【주민등록번호】	741011-1789814
【우편번호】	151-831
【주소】	서울특별시 관악구 봉천1동 715-143 플렉스빌 401호
【국적】	KR

【공개형태】 학회발표

【공개일자】 2002.11.21

【심사청구】 청구

【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인
황이남 (인)

【수수료】

【기본출원료】	20	면	29,000 원
【가산출원료】	8	면	8,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	20	항	749,000 원
【합계】	786,000 원		

【첨부서류】 1. 요약서·명세서(도면)_1통 2.공지예외적용대상(신규성상실의예외, 출원시의특례)규정을 적용받 기 위한 증명서류_1통

【요약서】

【요약】

본 발명은 컴퓨터 바이러스를 치료하는 방법으로서,

(1) 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 함수를 미리 저장된 감염되지 않은 함수와 비교하여 정상여부를 판단하는 단계; 및

(2) 정상인 함수를 이용하여 검색한 메모리의 프로세스 및 해당 파일을 대상으로 바이러스 감염여부의 진단 및 치료를 수행하는 단계를 포함하는 바이러스의 치료방법을 제공한다.

상기 구성에 의하면 바이러스에 감염될 수 있는 영역에 대한 정보, 특히 현재 메모리에 상주하는 프로세스를 빠짐없이 정확하게 검색하는 것이 가능하고, 메모리를 감염시키는 바이러스를 완벽하게 치료할 수 있다.

【대표도】

도 1

【명세서】

【발명의 명칭】

메모리를 감염시키는 바이러스의 치료방법, 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체 및 바이러스의 치료장치{Curative Method for Computer Virus Infecting Memory, Recording Medium Comprising Program Readable by Computer, and The Device}

【도면의 간단한 설명】

도 1은 본 발명에 따른 바이러스에 감염된 프로세스의 치료과정을 보여주는 모식도

도 2는 쓰레드 영역에 존재하는 바이러스를 진단/치료하는 과정을 보여주는 모식도

도 3은 본 발명의 제 1측면에 따른 바이러스에 감염된 프로세스의 치료과정을 나타내는 절차도

도 4는 본 발명의 제 2측면에 따른 바이러스에 감염된 프로세스의 치료과정을 나타내는 절차도

도 5는 본 발명의 제 3측면에 따른 바이러스에 감염된 프로세스의 치료과정을 나타내는 절차도

도 6은 본 발명의 일실시예로서 바이러스 치료장치를 나타내는 구성도

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<7> 본 발명은 컴퓨터에 저장된 파일 또는 실행중인 프로세스로부터 바이러스를 진단하고, 감염된 파일 또는 프로세스를 치료하는 방법, 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체 및 바이러스의 치료장치에 관한 것으로, 보다 상세하게는 바이러스에 감염될 수 있는 영역에 대한 정보, 특히 현재 메모리에 상주하는 프로세스를 빠짐없이 정확하게 검색하는 것이 가능하고, 메모리를 감염시키는 바이러스를 완벽하게 치료할 수 있는 방법, 기록매체 및 장치에 관한 것이다.

<8> 컴퓨터에서 프로그램이 실행될 때, 메모리에는 해당 프로그램의 프로세스가 상주하게 된다. 바이러스는 이와 같이 메모리에 상주하는 프로세스나 하드디스크 등의 기억장치에 저장되어 있는 프로그램 파일을 목표로 하며, 바이러스에 감염된 프로세스가 또 다른 프로세스나 파일을 감염시킴으로써 컴퓨터 바이러스가 전파되어 나간다.

<9> 종래에 메모리를 감염시키는 바이러스의 치료방법은 다음과 같다.

<10> 먼저, 메모리에 상주하고 있는 프로세스의 리스트를 검색하고, 기억장치(하드디스크 등)에서 그에 해당하는 파일이 바이러스에 감염되었는지를 진단한다. 진단결과, 파일이 바이러스에 감염되었을 경우에는 메모리에 상주하는 해당 프로세스를 킬(Kill) 시킨다. 그리고, 하드디스크에 저장되어 있는 해당 파일을 치료하여 다시 실행시킴으로써 정상적인 프로세스를 메모리에 상주시키는 방식이 사용된다.

<11> 그러나, 최근의 컴퓨터 바이러스는 백신이 바이러스에 감염될 수 있는 영역에 대한 정보를 검색할 때, 바이러스 자신을 먼저 실행시키도록 하여 검색결과에서 바이러스 자신을 해당 감염영역에 존재하지 않는 것처럼 누락시킨다.

<12> 위와 같은 경우에는 메모리에 상주하는 프로세스 리스트에서 바이러스에 감염된 프로세스를 누락시키는 것이다. 따라서, 종래의 방법으로는 백신 프로그램이 바이러스를 제대로 진단할 수 없는 문제점이 있다.

<13> 그리고, 종래 기술에 의해서는 파일은 감염시키지 않고 프로세스만 감염시키는 바이러스는 제대로 진단할 수 없으며, 프로세스에 종속되어 실행중인 쓰레드(Thread)만 감염된 경우에도 메모리가 바이러스에 감염되었는지를 전혀 진단할 수 없는 문제점이 있다.

【발명이 이루고자 하는 기술적 과제】

<14> 본 발명은 상기 종래 기술이 지니는 문제를 해결하기 위해 안출된 것으로,

<15> 본 발명의 목적은 바이러스에 감염될 수 있는 영역에 대한 정보로서, 특히, 현재 메모리에 상주하는 프로세스와 쓰레드를 빠짐없이 정확하게 검색하는 것이 가능하고, 메모리를 감염시키는 바이러스를 완벽하게 치료할 수 있는 방법을 제공하는데 있다.

<16> 본 발명의 다른 목적은 상기 바이러스를 치료할 수 있는 방법을 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하는데 있다.

<17> 본 발명의 또 다른 목적은 PC나 PDA, 핸드폰, 반도체, 기타 산업용 장비 등에 적용되는 하드웨어 장치를 포함하는 바이러스 치료장치를 제공하는데 있다.

【발명의 구성 및 작용】

<18> 용어 정의

<19> 바이러스: 사용자 몰래 컴퓨터 프로그램이나 실행가능한 부분을 변형하고, 여기에 자신 또는 이들의 일부 변형형태를 복사하는 일종의 프로그램으로서, 통상적으로 자기복제, 감염, 파괴작업 등을 수행하는 작은 프로그램을 의미한다. 본 발명의 기술적 사상이 적용될 수 있는 바이러스의 범위는 이와 같은 모든 형태의 바이러스 및 장래 발생 가능한 어떠한 형태의 바이러스도 모두 포함됨은 물론이다.

<20> 바이러스에 감염될 수 있는 영역: 통상적으로는 바이러스가 감염될 수 있는 영역인 기억장치로서, 주기억장치 및 보조기억장치를 모두 포함한다. 즉, 컴퓨터 바이러스가 일반적으로 감염시킬 수 있는 모든 대상을 의미하며, 메모리, 파일, 서비스, 레지스트리, TCP/I P 패킷 포트, 부트 등이 여기에 포함된다.

<21> 운영체제: 제한된 시스템의 자원들을 효율적으로 관리 및 운영함으로써, 사용자에게 편의성을 제공해 주는 인간과 기계간에 인터페이스 역할을 수행하는 프로그램을 의미한다. 이러한 운영체제에는 도스, 매킨토시, 윈도우즈, OS/2, 유닉스, 리눅스 등이 있다.

<22> 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 '함수': 운영체제가 제공하는 함수로서, 응용프로그램인터페이스(API), 시스템 호출(system call) 등을 포함한다.

<23> 프로세스: 하나의 독립된 프로그램의 실행 단위를 의미한다.

<24> 프로세스 킬: 프로세스의 종료, 즉, 메모리에서 해당 프로세스를 제거하는 것을 말한다.

<25> 상기 목적을 달성하기 위해, 제 1측면에 따른 본 발명은

<26> (1) 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 함수를 미리 저장된 감염되지 않은 함수와 비교하여 정상여부를 판단하는 단계; 및

<27> (2) 정상인 함수를 이용하여 검색한 메모리의 프로세스 및 해당 파일을 대상으로 바이러스 감염여부의 진단 및 치료를 수행하는 단계를 포함하는 바이러스의 치료방법을 제공한다.

<28> 제 2측면 및 제 3측면에 따른 본 발명은 상기 단계 (2)에 진단 및 치료대상으로 메모리의 쓰레드 영역이 더 추가된 바이러스의 치료방법을 제공한다.

<29> 또한 본 발명은,

<30> (1) 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 함수를 미리 저장된 감염되지 않은 함수와 비교하여 정상여부를 판단하는 단계; 및

<31> (2) 정상인 함수를 이용하여 검색한 메모리의 프로세스 및 해당 파일을 대상으로 바이러스 감염여부의 진단 및 치료를 수행하는 단계를 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

<32> 이하 본 발명의 내용을 대표적인 운영체제의 하나인 윈도우즈에 한정하여 도면과 함께 상세히 설명하고자 한다. 하지만, 본 발명의 기술적 사상은 윈도우즈에만 한정되지 않으며, 당업자라면 다른 유사 운영체제에도 동일하게 적용될 수 있을 것임을 용이하게 알 수 있을 것이다.

<33> 도 1은 본 발명에 따른 바이러스에 감염된 프로세스의 치료과정을 보여주는 모식도이다. 메모리(1)는 프로세스 리스트(2) 및 이와 매핑되는 프로세스 영역(3)을 포함한다. 부호 4는 기억장치를 나타낸다.

<34> 도 1에 제시된 치료과정을 예로서 본 발명의 내용을 설명하면 다음과 같다. 먼저, 메모리(1)에서 프로세스리스트(2) 및 엔트리 포인트(EP)를 찾아 각 프로세스가 감염되었는지 여부를 검색한다 (a). 만일 프로세스 B가 복구될 수 없을 정도로 손상된 경우 해당 프로세스를 킬(kill) 시킨다. 이때 바람직하게는 확인창을 통해 프로세스를 킬시키기 전에 이러한 사실을 확인시켜 준다. 프로세스를 킬시킨 후에 기억장치(4)에서 파일 B를 검색한다 (b). 파일 B에 대한 바이러스 진단 및 치료를 수행한 후, 파일 B를 재실행시킨다 (c). 상기 과정에 의해 메모리에는 바이러스가 치료된 프로세스 B가 상주한다 (d).

<35> 대부분의 백신프로그램은 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 API(Application Program Interface) 함수를 이용한다.

<36> 본 발명에 따른 바이러스 치료과정은 먼저, 각 함수의 코드가 정상인지를 확인하기 위해 바이러스에 감염되지 않은 API함수의 바이너리 코드를 운영체제별로 미리 저장하는 과정을 포함한다.

<37> 위 과정에 의하면 백신 프로그램이 미리 저장되어 있는 API 함수의 바이너리 코드와 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 API 함수의 바이너리 코드를 비교하여 정상여부를 확인할 수 있다.

<38> 백신 프로그램이 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하려고 사용하는 API 함수들을 예로 들면, 다음과 같은 것들이 있다.

<39> NTDLL.DLL::NtQuerySysteminformation

<40> NTDLL.DLL::NtResumeThread

<41> NTDLL.DLL::LdrGetDIIHandle

<42> KERNEL32.DLL::FindFirstFileExW

<43> KERNEL32.DLL::FindNextFileW

<44> ADVAPI32.DLL::Enum ServicesStatusA

<45> ADVAPI32.DLL::Enum ServicesStatusW

<46> ADVAPI32.DLL::RegEnumKeyExW

<47> ADVAPI32.DLL::RegEnumKeyW

<48> IPHLPAPI.DLL::GetTcpTableFromStack

<49> IPHLPAPI.DLL::GetUdpTableFromStack

<50> WinXP에 존재하는 NTDLL.DLL::NtQuerySysteminformation 함수가 바이러스에 감염되면 예로, 하기에서와 같은 코드의 변경이 수반된다.

<51>

```
B8(AC 00 00 00) mov eax,Ach              E9(6C 13 FD FF) jmp   0FFFD1371
BA 00 03 FE 7F mov  edx,7FFE0300h  ->    BA 00 03 FE 7F mov  edx,7FFE0300h
FF D2          call edx                  FF D2            call edx
C2 10 00       retn 10h                  C2 10 00         retn 10h
```

<52> 상기와 같이 API 함수의 코드가 변경되면, API 함수가 정상적으로 실행되기 전에 바이러스가 먼저 실행된다. 바이러스는 API 결과값에 자신이 존재하는 영역에 대한 정보가 출력되지 않도록 만든다. 따라서, API 함수가 제공하는 결과만을 가지고는 바이러스를 진단할 수 없다.

<53> 이러한 문제를 해결하기 위해 정상 API 함수의 코드를 미리 백신 프로그램 또는 기억장치(예를 들면, 하드 디스크) 등에 저장해 놓고, 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해서 사용할 API 함수를 미리 지정해 놓은 코드와 비교함으로써 해당 코드가 정상인지 아닌지를 확인할 수 있다.

<54> 비교과정에서 메모리에 상주하는 바이러스에 의해서 백신프로그램이 감염된다 하더라도, 이는 본 출원인에 의해 등록된 특허등록 제0370229호에 기재된 방법으로 치유가 가능하다.

<55> 코드를 비교한 결과, API 함수가 변경되지 않은 경우에는 API 함수로 검색한 메모리의 프로세스를 진단 및 치료하면 된다. 만일 메모리의 쓰레드 영역을 진단 및 치료하고자 하는 경우에는 프로세스 진단에 선행하여 쓰레드 영역을 먼저 진단하여도 된다.

<56> 또한 코드를 비교한 결과, API 함수의 코드가 변경되었으면 바이러스에 감염된 프로세스를 검색하는 것 자체가 불가능하므로, API 함수를 미리 저장해 놓은 코드로 복구시키고 프로세스 또는 쓰레드 영역을 진단 및 치료하면 된다.

<57> 이와 같은 과정을 통해 API 함수는 무결성을 가지게 된다. 상기 과정에서는 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해서 사용할 수 있는 API 함수 모두를 미리 저장해 놓았지만, 메모리에 상주하는 프로세스만을 검색하는 API 함수만 미리 저장해 놓을 수도 있다.

<58> 바이러스 중에는 메모리의 프로세스 영역만 감염시키고 파일영역은 감염시키지 않는 종류(예를 들면, 코드레드, 슬래머 등)들이 있다. 이런 종류의 바이러스 들은 메모리의 프로세스 영역을 진단하여 치료해야 한다.

<59> 먼저, API 함수를 이용하여 메모리에 상주하는 프로세스 리스트와 프로세스 각각의 시작 포인트(Entry Point; EP)를 찾는다. 이때 사용하는 API 함수로는 NTDLL.DLL::NtQuerySysteminformation 이나, NTDLL.DLL::LdrGetDIIHandle 이 있다.

<60> 다음으로, 메모리 페이지에서 해당 프로세스의 시작 포인트(EP)부터 바이러스에 감염되었는지를 진단한다. 바이러스에 감염된 프로세스를 백신이 치료할 수 있으면 바로 치료를 한다.

<61> 바이러스로 인하여 메모리의 프로세스가 심하게 손상된 경우에는 치료가 불가능하므로, 메모리에 상주하는 프로세스를 킬시킨다. 예를 들면, 메모리에 A, B, C 프로세스가 상주하고 있고, B 프로세스가 바이러스에 감염되어 치료가 불가능 하다면, 메모리에 상주하는 B 프로세스를 킬시킨다.(도 1 참조)

<62> 메모리에 상주하는 B 프로세스를 킬시키기 전에 B 프로세스를 킬시키겠다는 메시지를 사용자에게 디스플레이 시켜주는 것이 바람직하다. 상기 메시지를 디스플레이 하는 이유는 사용자가 이미 작업 중인 B 프로세스가 백신 프로그램에 의하여 임의적으로 종료되어 작업 중인 내용이 사라지는 것을 방지하고, 사용자가 상기 메시지를 보고 해당 작업을 저장할 수 있는 시간을 주기 위한 것이다.

<63> 사용자가 확인 메시지를 클릭하면 B 프로세스는 메모리에서 킬처리된다.

<64> 그리고 기억장치(예로, 하드디스크)에서 프로세스에 해당하는 파일을 검색한다. 도 1을 예를 들면, B 프로세스에 해당하는 파일을 기억장치에서 검색한다.

<65> 하드디스크에 해당 파일이 검색되지 않으면 백신프로그램을 종료한다.

<66> 기억장치에서 프로세스에 해당하는 파일을 검색하여 파일이 존재하는 경우, 해당 파일이 감염되었는지 진단하여 치료를 한다. 그리고, 필요에 따라서는 메모리 쓰레드 영

역을 진단 및 치료하는 과정을 더 수행하는 것이 바람직하다. 이에 관한 상세한 내용은 별도로 후술하기로 한다.

<67> 기억 장치에 저장되어 있는 파일을 치료한 경우에는, 해당 파일을 재실행시키는 것이 바람직하다. 파일을 재실행시키면 메모리에는 바이러스에 감염되지 않은 B 프로세스가 상주하게 되므로 바이러스를 완전히 치료하게 된다. 여기서, B 프로세스를 다시 상주시키는 이유는 운영체제가 사용하는 프로세스의 경우, 해당 프로세스가 치료과정에서 킬 처리되면 운영체제가 원활하게 작동한다는 보장을 받을 수 없기 때문이다.

<68> 바이러스에 의해 심하게 손상된 프로세스는 치료로 이미 킬처리되었기 때문에, 이때 기억장치에 존재하는 파일은 감염되지 않게 된다.

<69> 메모리에는 프로세스 영역과는 별개로 쓰레드 영역이 따로 존재한다. 쓰레드 영역을 감염시키는 바이러스들(예를 들면, Elkern 바이러스 등)은 대부분 프로세스의 쓰레드 영역에 바이러스에 감염된 쓰레드를 추가시킴으로써 감염시킨다.

<70> 따라서 추가된 쓰레드를 킬시키면 사용자가 사용 중인 프로세스에는 전혀 영향을 미치지 않고 바이러스를 치료할 수 있다.

<71> 도 2는 쓰레드 영역에 존재하는 바이러스를 진단/치료하는 과정을 보여주는 모식도이다. 먼저 쓰레드 영역에 존재하는 바이러스를 진단/치료하기 위해서는 메모리에 상주하고 있는 프로세스 각각에 대한 쓰레드 리스트와 각 쓰레드에 대한 시작 포인트(EP)를 찾아야 한다. 여기에서도 상기에서와 마찬가지로 API 함수(예를

들면, NTDLL.DLL::NtResumeThread)를 이용하여 쓰레드 리스트와 각 쓰레드의 시작 포인트를 찾아낼 수 있다.

<72> 그런 다음, 메모리 페이지에서 해당 쓰레드의 시작 포인트부터 바이러스에 감염되었는지를 진단한다. 진단결과 바이러스에 감염된 쓰레드(색칠한 부위)가 있으면, 해당 쓰레드를 메모리 영역에서 킬시킨다. 이에 의해 사용자가 사용중인 프로세스를 킬시키지 않고도 바이러스를 치료할 수 있다.

<73> 이하 본 발명의 내용을 도 3 내지 도 5의 바람직한 실시예를 통해 보다 상세하게 설명하고자 한다. 다만 이들 실시예는 본 발명의 내용을 이해하기 위해 제시되는 것일 뿐 본 발명의 권리범위가 이들 실시예에 한정되어지는 것으로 해석되어져서는 아니된다.

<74> 도 3은 제 1측면에 따른 본 발명의 바람직한 실시예를 나타낸다. 바이러스에 감염되지 않은 정상적인 API 함수의 바이너리 코드는 백신 프로그램 또는 기억장치(예를 들면, 하드 디스크)에 미리 저장되어 있다. 단계 301에서는 상기 저장된 API 함수와 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 API 함수의 바이너리 코드를 상호 비교한다. 단계 302에서 코드가 동일하다고 판단된 경우에는 단계 304로 넘어가서 프로세스의 바이러스 감염여부를 진단하고, 코드가 변경된 것으로 판단된 경우에는 API 함수를 미리 저장해 놓은 코드로 복구시키고(단계 303), 단계 304로 들어간다.

<75> 단계 304에서 메모리에 상주하는 프로세스가 바이러스에 감염되었는지의 여

부를 진단하고, 단계 305에서 감염된 프로세스가 존재하는 것으로 판단된 경우 단계 306에서 치료가능여부를 판단한다. 단계 306에서 치료가능하다고 판단된 경우에는 단계 311에서 해당 프로세스를 치료하고, 단계 308에서 해당 파일을 기억장치에서 검색한다. 치료가 불가능하다고 판단된 경우에는 단계 307에서 해당 프로세스를 킬시킨 후, 단계 308에서 해당 파일을 기억장치에서 검색한다. 만일 단계 309에서 해당 파일이 기억장치에 존재하는 경우 단계 310에서 해당 파일을 진단 및 치료하고 이를 재실행시킨다. 이와는 달리 단계 309에서 해당 파일이 기억장치에 존재하지 않는 것으로 판단된 경우 바로 종료한다.

<76> 상기 과정에 의하면 API 함수의 무결성을 보장받을 수 있어 메모리를 감염시키는 바이러스를 완벽하게 치료할 수 있다.

<77> 도 4는 제 2측면에 따른 본 발명의 바람직한 실시예를 나타낸다. 상기 과정은 도 3의 제 1측면에 따른 실시예와 비교할 때 쓰레드 영역을 진단 및 치료하는 과정을 둔 점에서 상이하다. 상기 실시예에서는 메모리의 쓰레드 영역에 대한 진단 및 치료과정은 프로세스 또는/및 파일의 진단 및 치료과정이 종료한 후에 수행된다. (단계 412)

<78> 도 5는 제 3측면에 따른 본 발명의 바람직한 실시예로서, 메모리의 쓰레드 영역에 대한 진단 및 치료과정이 프로세스의 진단과정 이전에 수행되는 점에서 제 2측면에 따른 실시예와 상이하다. 즉, 단계 504에서 메모리의 쓰레드 영역을 진단 및 치료한 다음, 단계 505에서 메모리에 상주하는 프로세스가 바이러스에 감염되었는지의 여부를 진단하고,

단계 506에서 감염된 프로세스가 존재하는 것으로 판단된 경우 단계 507에서 치료가능여부를 판단한다. 단계 507에서 치료가능하다고 판단된 경우에는 단계 511에서 해당 프로세스를 치료하고, 단계 509에서 해당 파일을 기억장치에서 검색한다. 치료가 불가능하다고 판단된 경우에는 단계 508에서 해당 프로세스를 킬시킨 후, 단계 509에서 해당 파일을 기억장치에서 검색한다. 만일 단계 510에서 해당 파일이 기억장치에 존재하는 경우 단계 512에서 해당 파일을 진단 및 치료하고 이를 재실행시킨다. 이와는 달리 단계 510에서 해당 파일이 기억장치에 존재하지 않는 것으로 판단된 경우 바로 종료한다.

<79> 상기 제 2 및 제 3측면에 따른 실시예에서 설명된 바와 같이 쓰레드 영역의 진단 및 치료는 프로세스의 진단 및 치료과정의 이전 또는 이후의 어느 단계에서든 수행이 가능하다.

<80> 이상과 같이 설명한 본 발명에 따른 바이러스 치료과정은 컴퓨터 시스템에서 실행할 수 있는 프로그램으로 작성될 수 있다. 또한, 이러한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체로부터 해당 프로그램을 읽어들여 범용 디지털 컴퓨터 시스템에서 실행될 수 있다. 이러한 기록 매체에는 마그네틱 저장 매체(예를 들면, 롬, 플로피디스크, 하드디스크 등), 광학적 판독 매체(예를 들면, 씨디롬, 디브이디 등) 및 캐리어 웨이브(예를 들면, 인터넷을 통한 전송)와 같은 매체가 포함된다.

<81> 그러나, 본 발명의 실시가능한 형태는 이에 한정되지 않으며, PC나 PDA, 핸드폰, 반도체, 기타 산업용 장비 등에 적용되는 하드웨어 장치(바이러스 치료장치)로도 구현이 가능하다. 이와 같이 구현될 때, 바이러스 치료장치는 도 6에서와 같이 복원수단, 프로세스 치료수단 및 파일 치료수단을 포함할 수 있다.

<82> 복원수단은 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위한 API 함수의 바이너리 코드를 미리 저장된 감염되지 않은 API 함수의 바이너리 코드와 비교하여, 변경된 API 함수의 바이너리 코드를 원래의 바이너리 코드로 복구한다.

<83> 이 때, 바이러스 치료장치는 감염되지 않은 API 함수의 바이너리 코드를 저장한 원본 기억장치를 더 포함할 수 있으며, API 함수의 바이너리 코드는 운영체제별로 저장되어 있는 것이 바람직하다.

<84> 프로세스 치료수단은 API 함수를 이용하여 프로세스 리스트와 프로세스 각각의 시작 포인트(Entry Point; EP)를 찾는다. 그리고, 메모리 페이지에서 해당 프로세스의 시작 포인트(EP)부터 바이러스에 감염되었는지를 진단하여 치료한다.

<85> 치료가 불가능하면 메모리에 상주하는 프로세스를 킬시킨다. 이 때에도, 위에서 설명한 바와 마찬가지로 프로세스를 킬시키겠다는 메시지를 사용자에게 미리 디스플레이 시켜주는 것이 바람직하다.

<86> 파일 치료수단은 프로세스 치료수단에서 검색된 프로세스에 해당하는 파일을 찾아서 바이러스에 감염되었는지를 진단 및 치료하고, 해당 파일을 재실행시킨다.

<87> 한편, 바이러스 치료장치는 쓰레드를 치료하는 쓰레드 치료수단을 더 포함할 수 있다. 쓰레드 치료수단은 API 함수를 이용하여 메모리에 상주하고 있는 프로세스 각각에 대한 쓰레드 리스트와 각 쓰레드에 대한 시작 포인트(EP)를 찾아낸다. 그리고, 메모리 페이지에서 해당 쓰레드의 시작 포인트부터 쓰레드가 바이러스에 감염되었는지를 진단 및 치료한다.

<88> 쓰레드 치료수단은 파일 치료수단이 파일을 치료한 후에 쓰레드를 검색하여 치료할 수도 있고, 프로세스 치료수단이 API 함수를 이용하여 메모리에 상주하는 프로세스를 검색하기 전에 쓰레드를 검색하여 치료할 수도 있다.

【발명의 효과】

<89> 상기 본 발명의 구성에 의하면 바이러스에 감염될 수 있는 영역에 대한 정보, 특히 현재 메모리에 상주하는 프로세스를 빠짐없이 정확하게 검색하는 것이 가능하고, 메모리를 감염시키는 바이러스를 완벽하게 치료할 수 있다.

【특허청구범위】

【청구항 1】

컴퓨터 바이러스의 치료방법에 있어서,

(1) 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 함수를 미리 저장된 감염되지 않은 함수와 비교하여 정상여부를 판단하는 단계; 및

(2) 정상인 함수를 이용하여 검색한 메모리의 프로세스 및 해당 파일을 대상으로 바이러스 감염여부의 진단 및 치료를 수행하는 단계를 포함함을 특징으로 하는 바이러스의 치료방법

【청구항 2】

제 1항에 있어서, 단계 (2)의 정상인 함수는

당해 사용할 함수가 변경되지 않은 경우 당해 함수이며, 변경된 경우에는 미리 저장해 놓은 함수로 복구된 함수임을 특징으로 하는 바이러스의 치료방법

【청구항 3】

제 1항에 있어서, 단계 (2)는

메모리에 상주하는 프로세스를 진단하는 단계;

상기 프로세스를 진단하여 치료가 가능하면 치료하고, 치료가 불가능하면 상기 프로세스를 종료시키는 단계; 및

해당 프로세스의 파일을 검색하여 진단 및 치료하고 재실행시키는 단계를 포함하는 것을 특징으로 하는 바이러스의 치료방법

【청구항 4】

제 1항에 있어서,

단계 (2)에 진단 및 치료대상으로 메모리의 쓰레드 영역이 더 추가된 것을 특징으로 하는 바이러스의 치료방법

【청구항 5】

제 4항에 있어서, 단계 (2)는

메모리에 상주하는 프로세스를 진단하는 단계;

상기 프로세스를 진단하여 치료가 가능하면 치료하고, 치료가 불가능하면 상기 프로세스를 종료시키는 단계;

해당 프로세스의 파일을 검색하여 진단 및 치료하고 재실행시키는 단계; 및

메모리의 쓰레드 영역을 진단 및 치료하는 단계를 포함하는 것을 특징으로 하는 바이러스의 치료방법

【청구항 6】

제 4항에 있어서, 단계 (2)는

메모리의 쓰레드 영역을 진단 및 치료하는 단계;

메모리에 상주하는 프로세스를 진단하는 단계;

상기 프로세스를 진단하여 치료가 가능하면 치료하고, 치료가 불가능하면 상기 프로세스를 종료시키는 단계;

해당 프로세스의 파일을 검색하여 진단 및 치료하고 재실행시키는 단계를 포함하는 것을 특징으로 하는 바이러스의 치료방법

【청구항 7】

제 1항에 있어서,

함수는 도스, 매킨토시, 윈도우즈, OS/2, 유닉스, 리눅스에서 제공됨을 특징으로 하는 바이러스의 치료방법

【청구항 8】

제 1항에 있어서,

함수는 응용프로그램인터페이스(API) 함수 또는 시스템 콜 임을 특징으로 하는 바이러스의 치료방법

【청구항 9】

(1) 바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 함수를 미리 저장된 감염되지 않은 함수와 비교하여 정상여부를 판단하는 단계; 및

(2) 정상인 함수를 이용하여 검색한 메모리의 프로세스 및 해당 파일을 대상으로 바이러스 감염여부의 진단 및 치료를 수행하는 단계를 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【청구항 10】

제 9항에 있어서, 단계 (2)의 정상인 함수는

당해 사용할 함수가 변경되지 않은 경우 당해 함수이며, 변경된 경우에는 미리 저장해 놓은 함수로 복구된 함수임을 특징으로 하는 기록매체

【청구항 11】

제 9항에 있어서, 단계 (2)는

메모리에 상주하는 프로세스를 진단하는 단계;

상기 프로세스를 진단하여 치료가 가능하면 치료하고, 치료가 불가능하면 상기 프로세스를 종료시키는 단계; 및

해당 프로세스의 파일을 검색하여 진단 및 치료하고 재실행시키는 단계를 포함하는 것을 특징으로 하는 기록매체

【청구항 12】

제 9항에 있어서,

단계 (2)에 진단 및 치료대상으로 메모리의 쓰레드 영역이 더 추가된 것을 특징으로 하는 기록매체

【청구항 13】

제 12항에 있어서, 단계 (2)는

메모리에 상주하는 프로세스를 진단하는 단계;

상기 프로세스를 진단하여 치료가 가능하면 치료하고, 치료가 불가능하면 상기 프로세스를 종료시키는 단계;

해당 프로세스의 파일을 검색하여 진단 및 치료하고 재실행시키는 단계; 및

메모리의 쓰레드 영역을 진단 및 치료하는 단계를 포함하는 것을 특징으로 하는 기록매체

【청구항 14】

제 12항에 있어서, 단계 (2)는

메모리의 쓰레드 영역을 진단 및 치료하는 단계;

메모리에 상주하는 프로세스를 진단하는 단계;

상기 프로세스를 진단하여 치료가 가능하면 치료하고, 치료가 불가능하면 상기 프로세스를 종료시키는 단계;

해당 프로세스의 파일을 검색하여 진단 및 치료하고 재실행시키는 단계를 포함하는 것을 특징으로 하는 기록매체

【청구항 15】

제 9항에 있어서,

함수는 응용프로그램인터페이스(API) 함수 또는 시스템 콜 임을 특징으로 하는 기록매체

【청구항 16】

바이러스에 감염될 수 있는 영역에 대한 정보를 검색하기 위해 사용할 함수가 변경된 경우 이를 복원하기 위한 복원수단;

정상인 함수를 이용하여 프로세스 리스트와 프로세스 각각의 시작 포인트를 찾고, 메모리 페이지에서 해당 프로세스의 시작 포인트부터 바이러스에 감염되었는지를 진단하여 치료하는 과정을 수행하는 프로세스 치료수단; 및

상기 프로세스 치료수단에서 검색된 프로세스에 해당하는 파일을 검색하여 바이러스에 감염되었는지를 진단 및 치료하고, 해당 파일을 재실행시키는 파일치료수단을 포함하는 바이러스 치료장치

【청구항 17】

제 16항에 있어서,

메모리의 쓰레드 영역을 진단 및 치료하는 쓰레드 치료수단이 더 추가됨을 특징으로 하는 바이러스의 치료장치

【청구항 18】

제 16항에 있어서,

함수는 도스, 매킨토시, 윈도우즈, OS/2, 유닉스, 리눅스에서 제공됨을 특징으로 하는 바이러스의 치료장치

【청구항 19】

제 16항에 있어서,

함수는 응용프로그램인터페이스(API) 함수 또는 시스템 콜 임을 특징으로 하는 바이러스의 치료장치

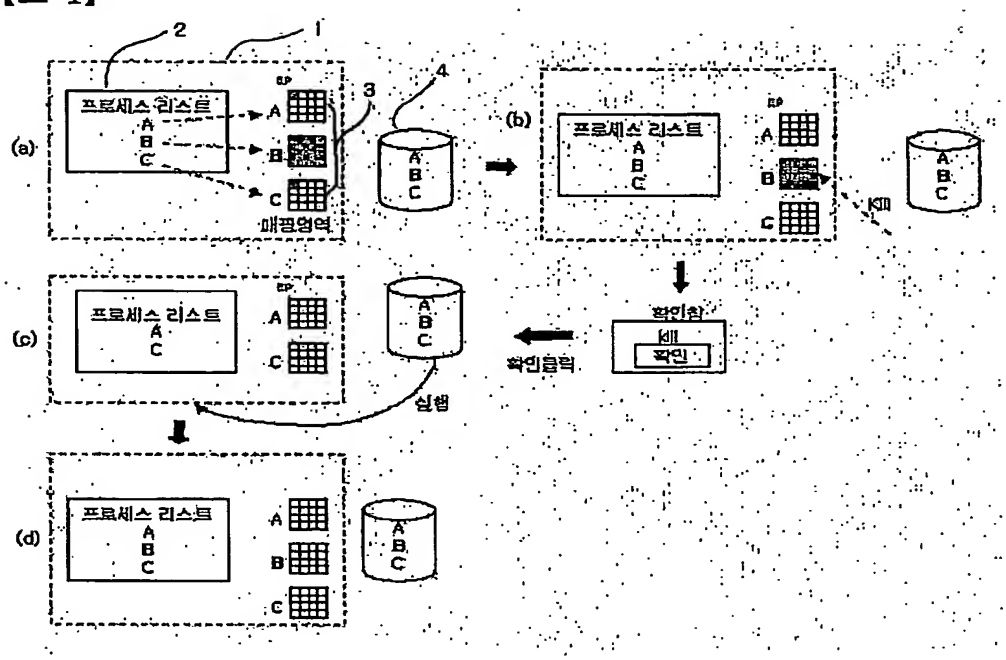
【청구항 20】

제 16항에 있어서,

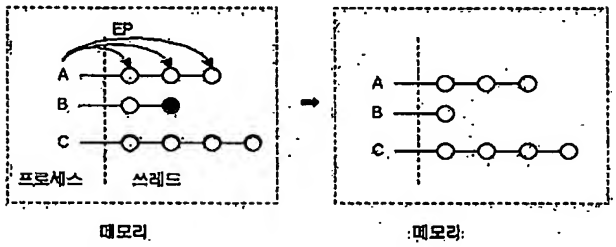
상기 바이러스 치료장치는 PC, PDA, 핸드폰, 반도체를 포함하는 산업용 장비 에 적용되는 하드웨어 장치임을 특징으로 하는 바이러스의 치료장치

【도면】

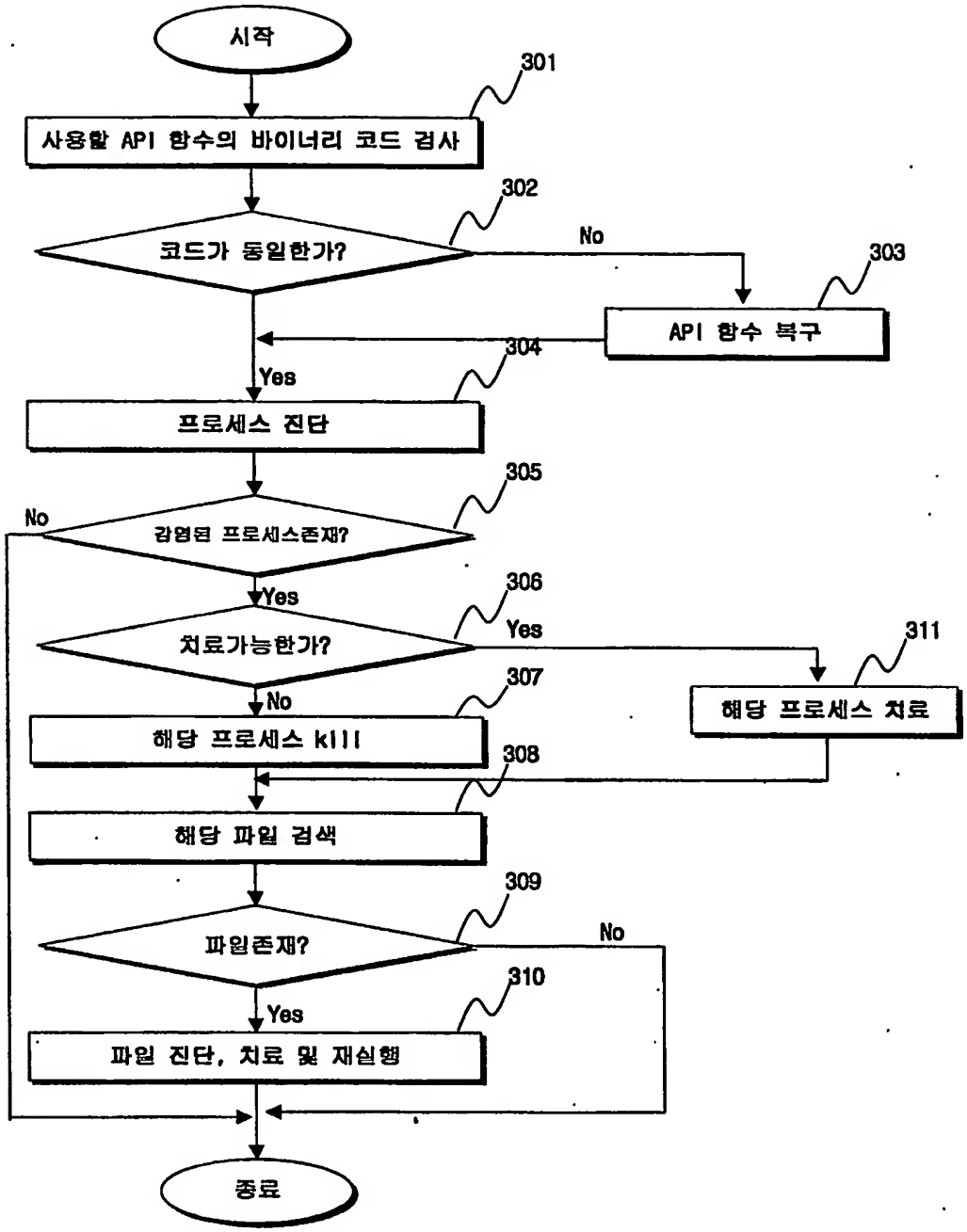
【도 1】



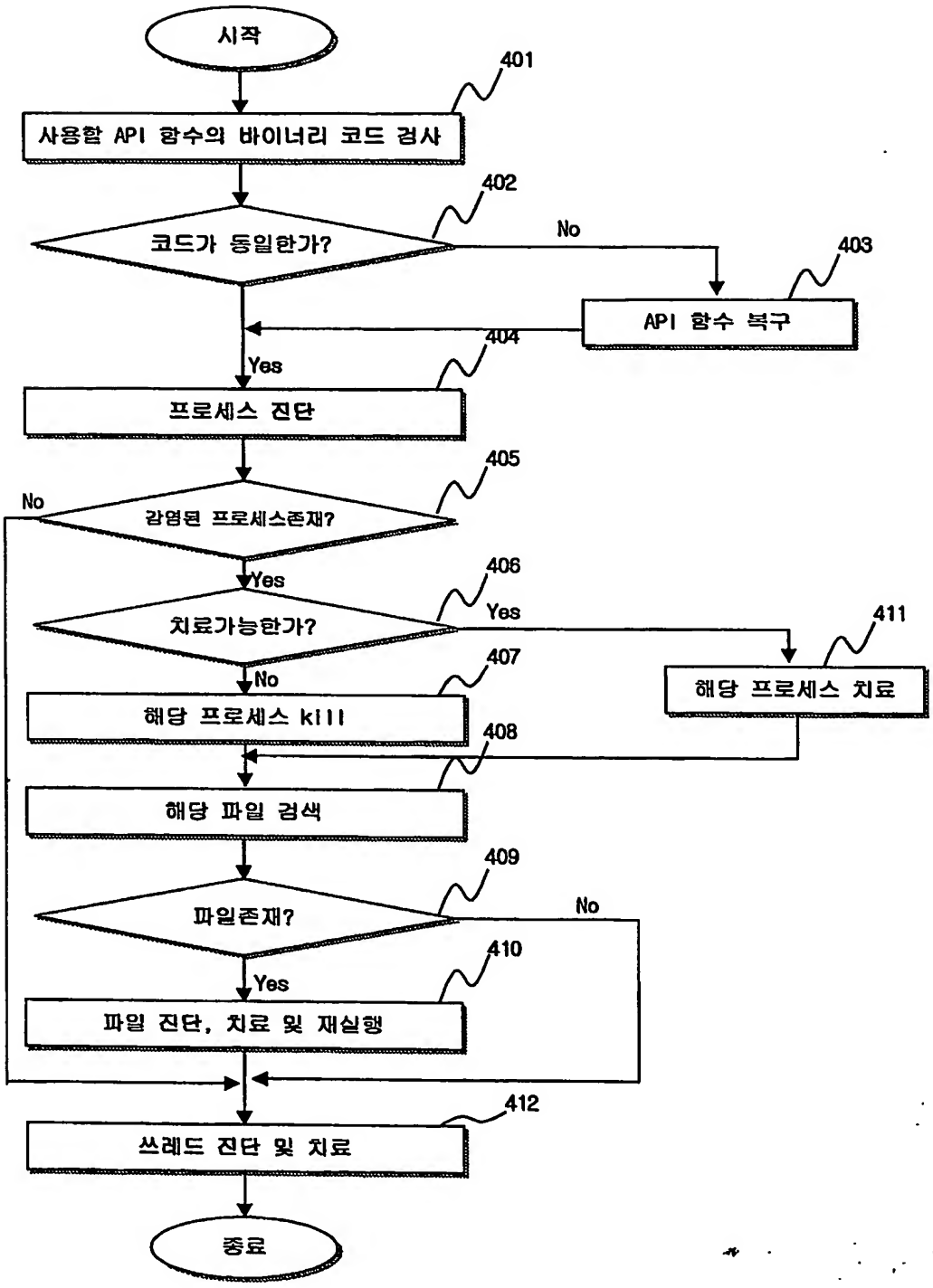
【도 2】



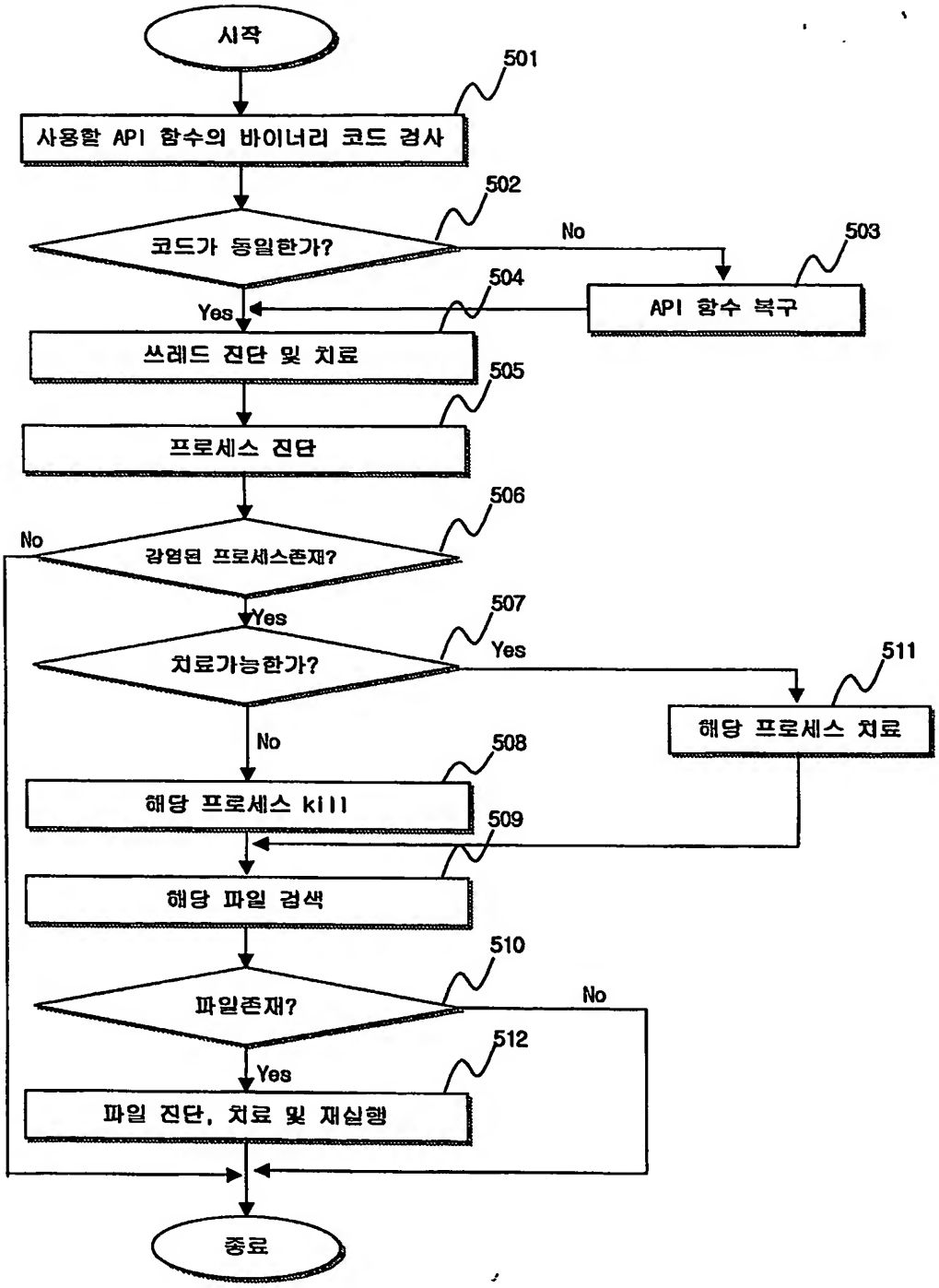
【도 3】


시작
301
사용할 API 함수의 바이너리 코드 검사
302
코드가 동일한가?
No
303
API 함수 복구
Yes
304
프로세스 진단
305
No
감염된 프로세스존재?
Yes
306
처료가능한가?
Yes
311
해당 프로세스 처료
No
307
해당 프로세스 kill
308
해당 파일 검색
309
파일존재?
No
310
Yes
파일 진단, 치료 및 재실행
종료

【도 4】


시작
401
사용할 API 함수의 바이너리 코드 검사
402
코드가 동일한가?
No
403
API 함수 복구
Yes
404
프로세스 진단
405
No
감염된 프로세스존재?
Yes
406
치료가능한가?
Yes
411
해당 프로세스 치료
No
407
해당 프로세스 kill
408
해당 파일 검색
409
파일존재?
No
410
Yes
파일 진단, 치료 및 재실행
412
쓰레드 진단 및 치료
종료

【도 5】


시작
501
사용할 API 함수의 바이너리 코드 검사
502
코드가 동일한가?
No
503
API 함수 복구
Yes
504
쓰레드 진단 및 치료
505
프로세스 진단
506
No
감염된 프로세스존재?
Yes
507
치료가능한가?
Yes
511
해당 프로세스 치료
No
508
해당 프로세스 kill
509
해당 파일 검색
510
파일존재?
No
Yes
512
파일 진단, 치료 및 재실행
종료

【도 6】